**Figure 1**

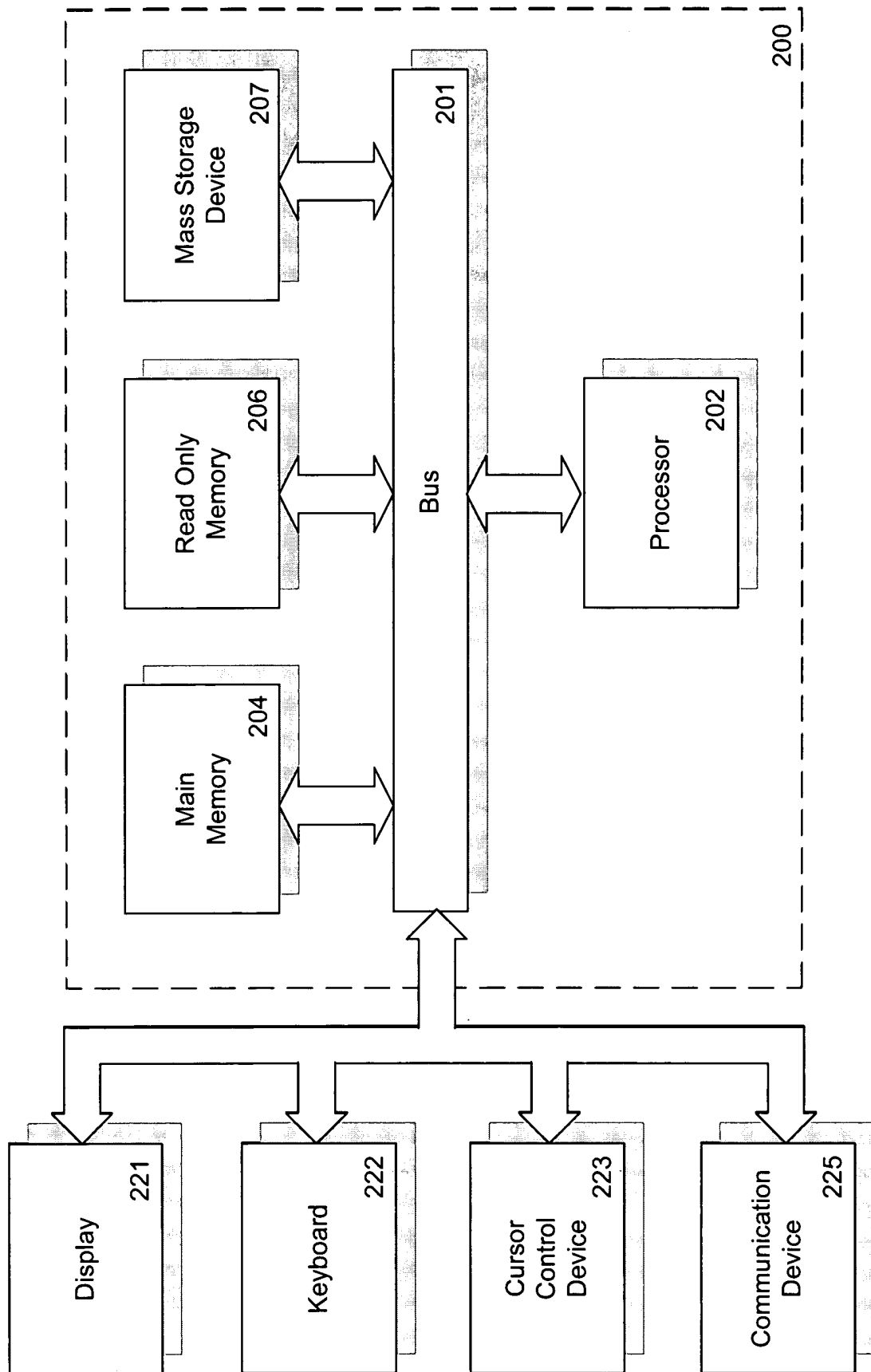


Figure 2

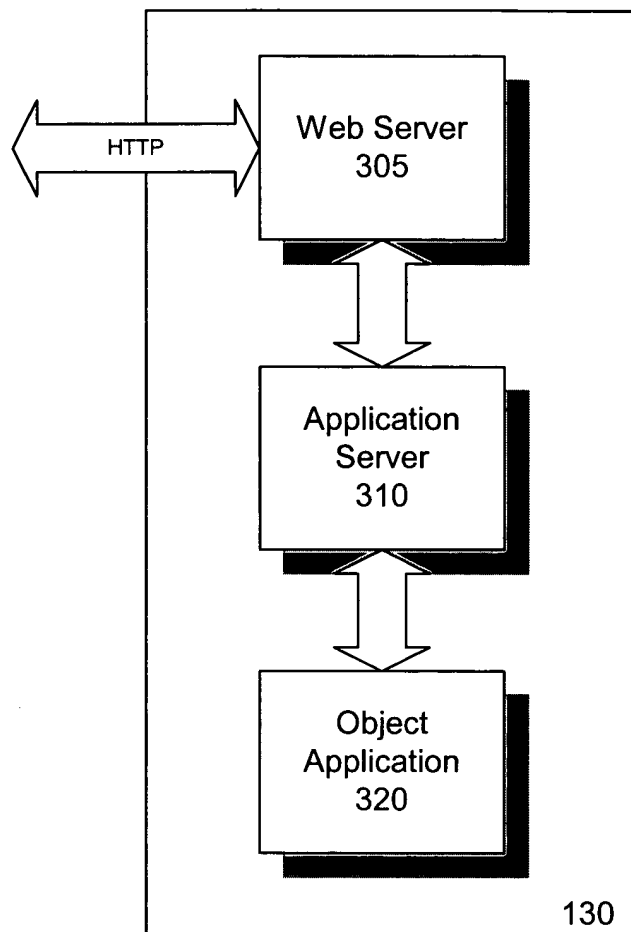


Figure 3

0966219-082701

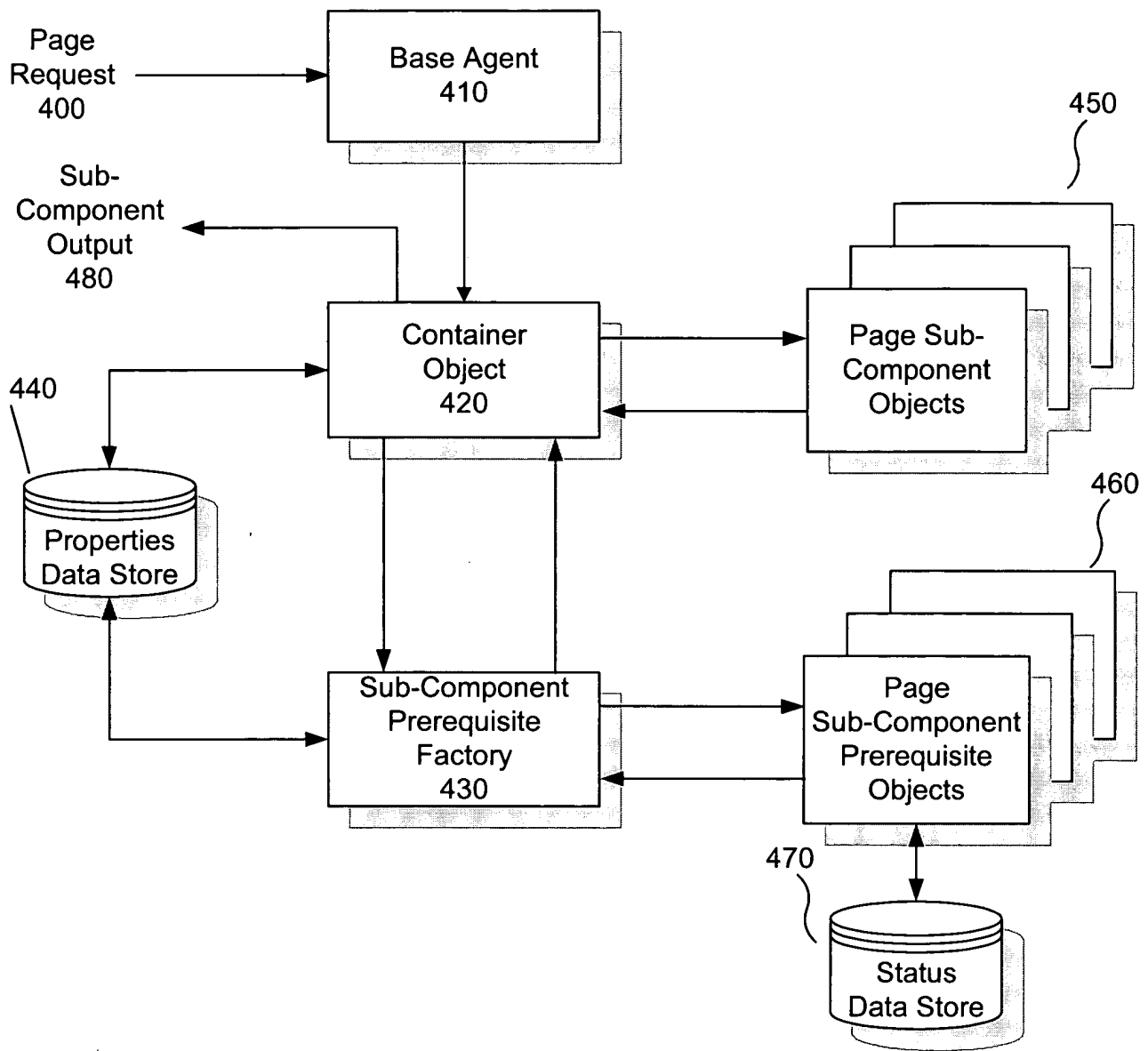


Figure 4

09826219-082701

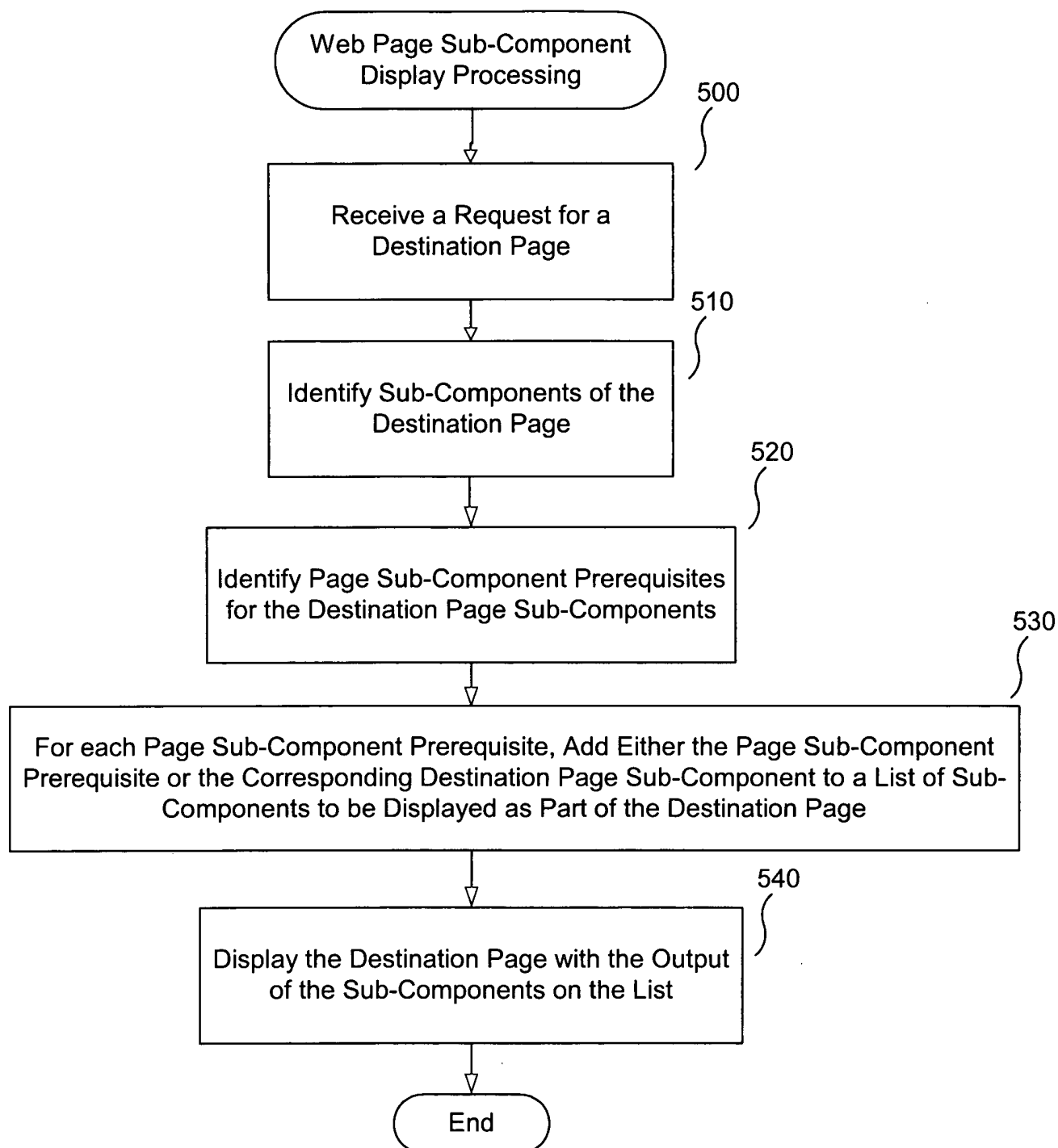


Figure 5A

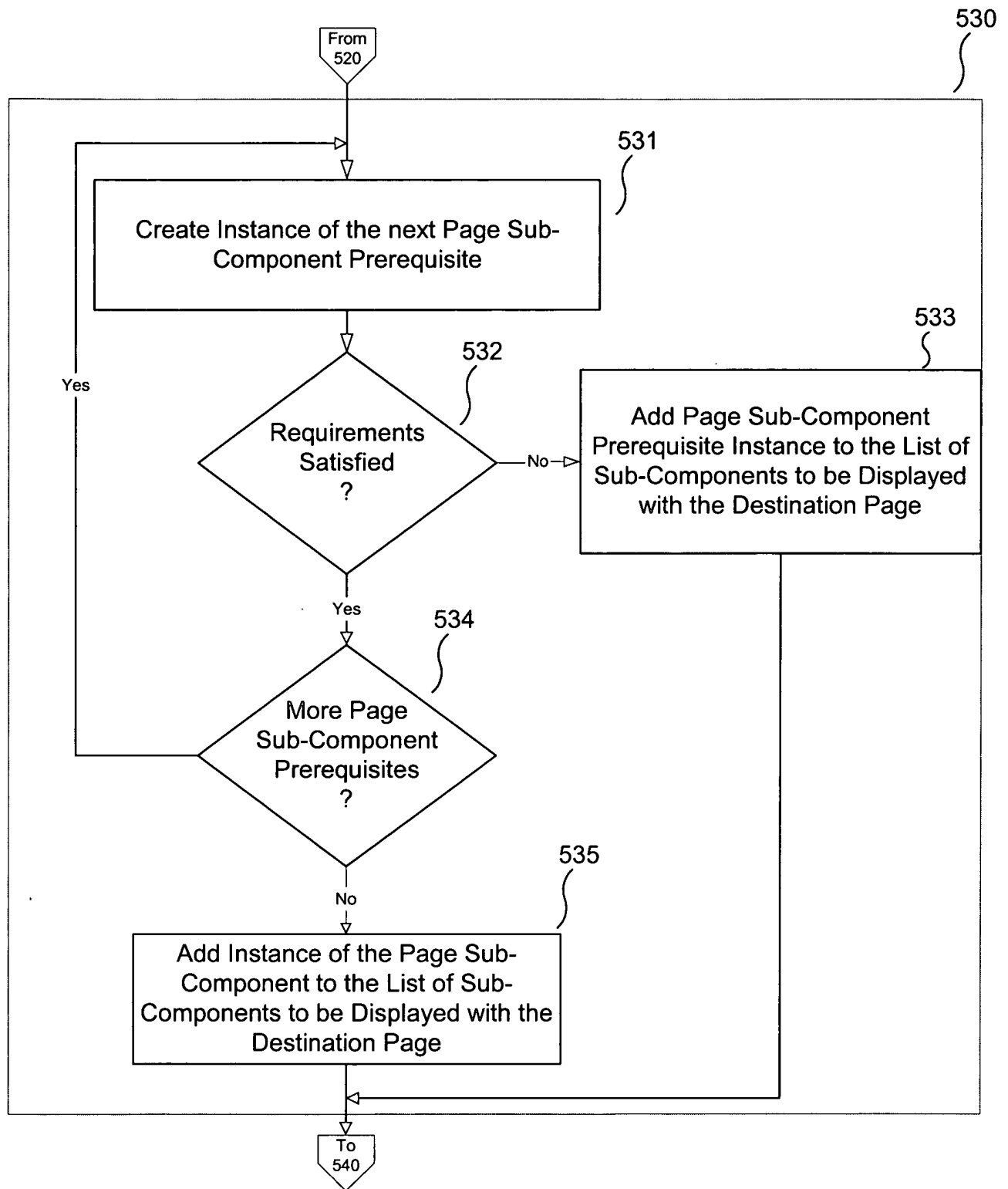


Figure 5B

[illegible]

Figure 6

Web Page Display Processing

Receive a Request to
Display a Destination Page

Identify Prerequisite Pages
Associated with the Destination Page

Sequentially Display Prerequisite Pages whose
Requirements Have not been Satisfied

Display the Destination Page

End

710

720

730

740

Figure 7A

0986219-082701

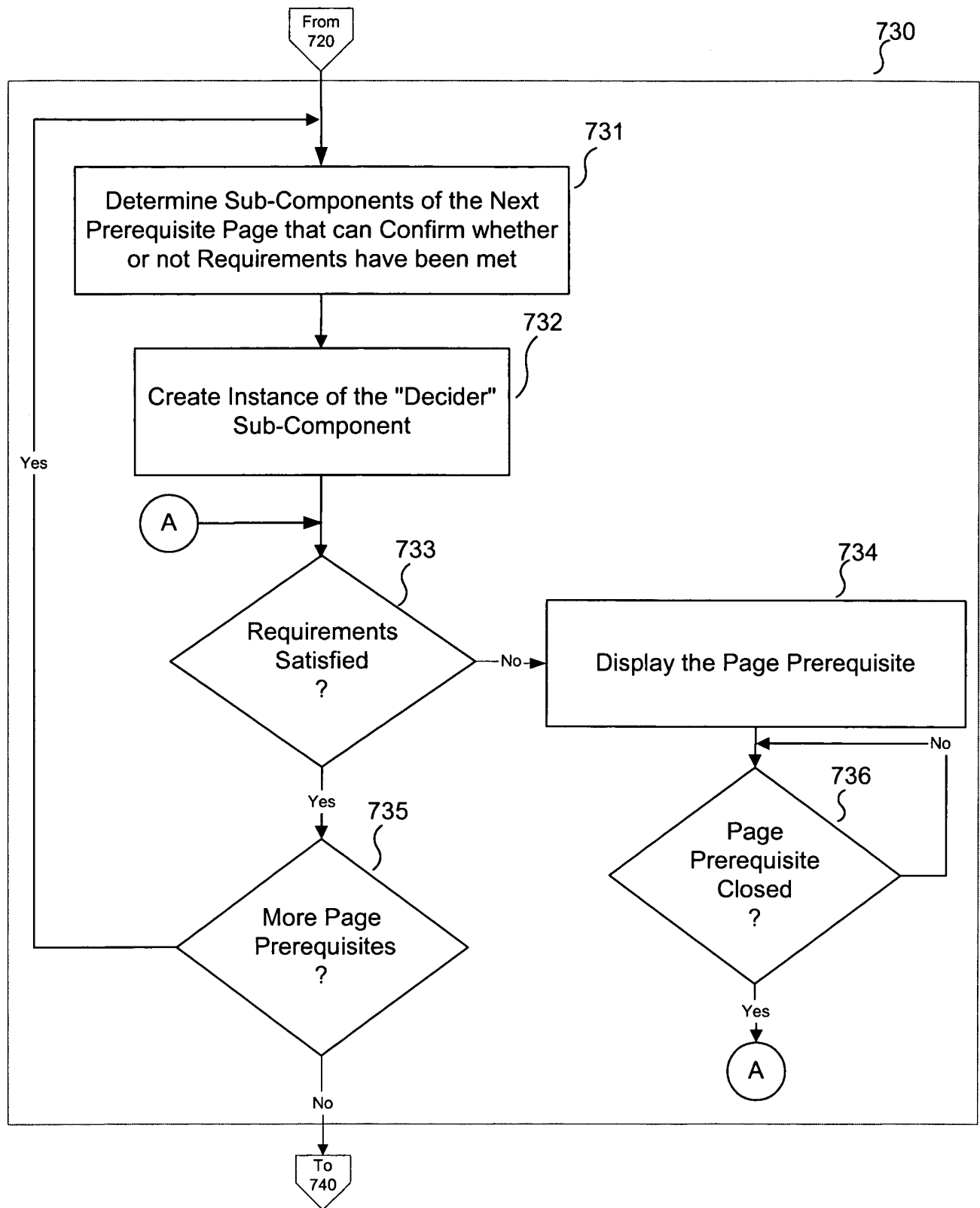


Figure 7B

810
<Attribute> = <Value>
820
prompt1 = please enter your name

Figure 8 (Prior Art)

910
<Object>.<Attribute> = <Value>
920
Obj1.attr1 = val11
930
Obj1.attr2 = val12
940
Obj2.attr1 = val21
950
Obj2.attr3 = val23
960
Trading.prompt1 = please enter your name

Figure 9

<Object>.extends = <anotherObject> 1010

Obj1.attr1 = val11 1020

Obj1.attr2 = val12 1030

Obj2.extends=Obj1 1040

Obj2.attr2 = val22 1050

Obj2.attr3 = val23 1060

Figure 10

<Context>.<Object>.<Attribute> = <Value> 1110

Obj1.attr1 = val11 1120

Obj1.attr2 = val12 1130

Page1.Obj1.attr2 = val112 1140

Figure 11

<Context>.extends = <anotherContext> 1210

Obj1.attr1 = val11 1220

Obj1.attr2 = val12 1230

Page1.Obj1.attr2 = val112 1240

Page2.extends = Page1 1250

Page2.Obj1.attr2 = val212 1260

Figure 12

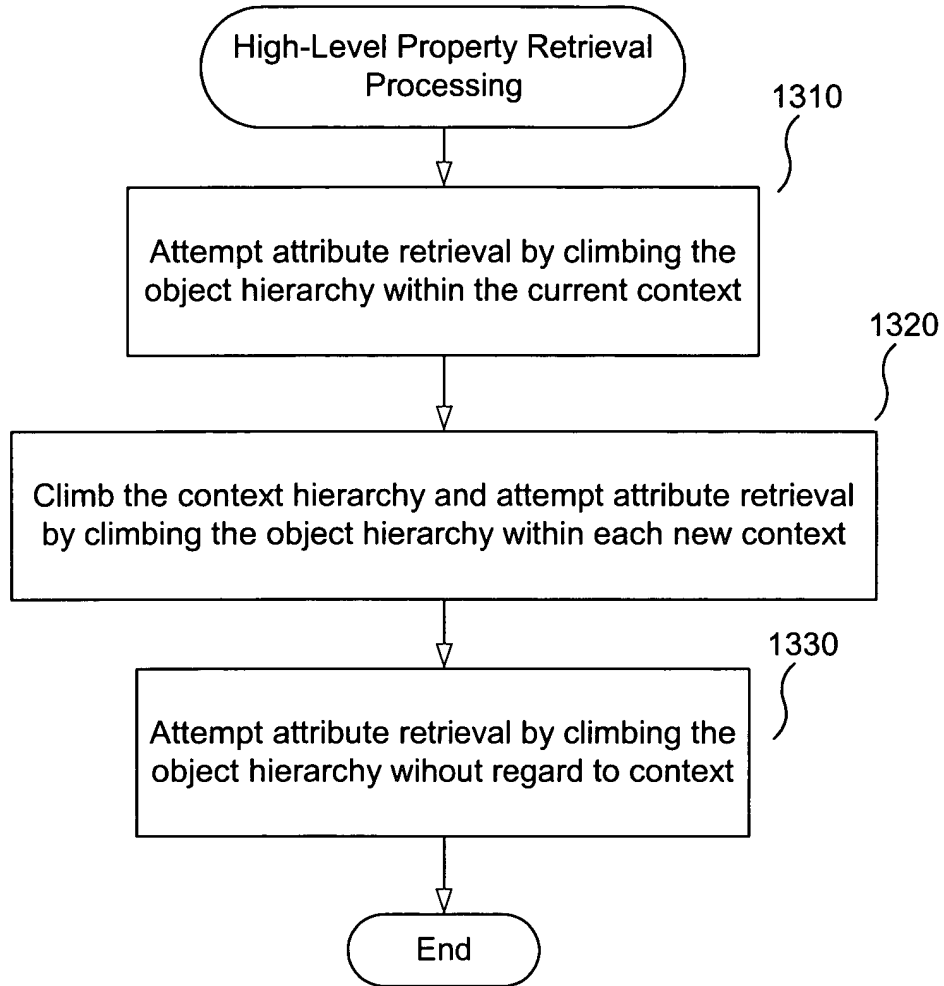


Figure 13

Page2.extends=Page1
Page3.extends=Page2

Obj2.extends=Obj1
Obj3.extends=Obj2

Figure 14A

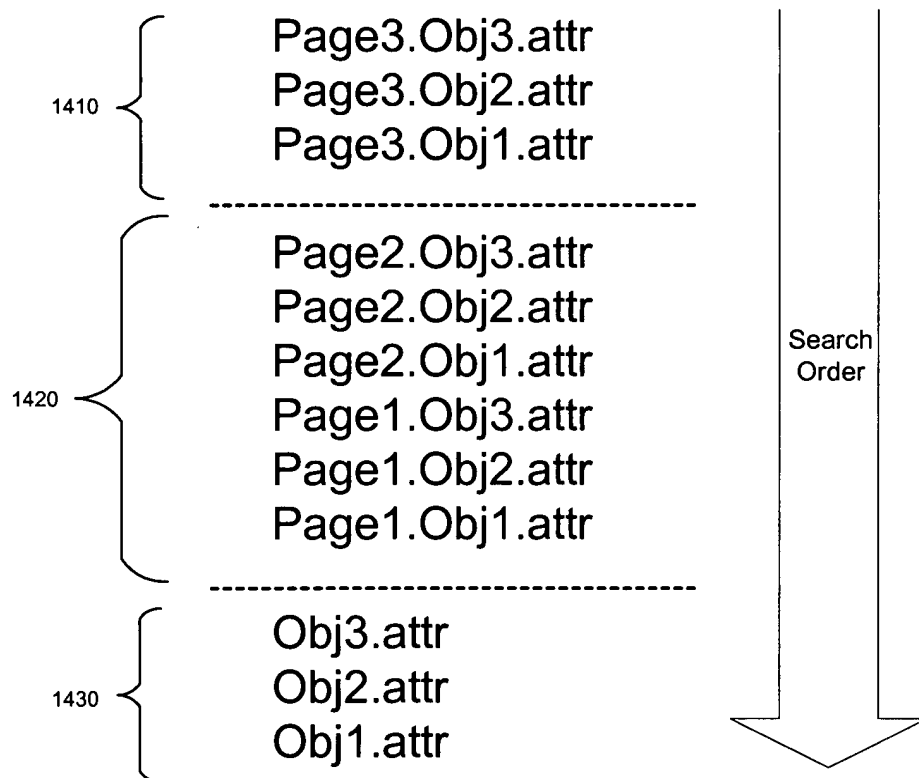


Figure 14B

10/22/00 6:29:28 PM

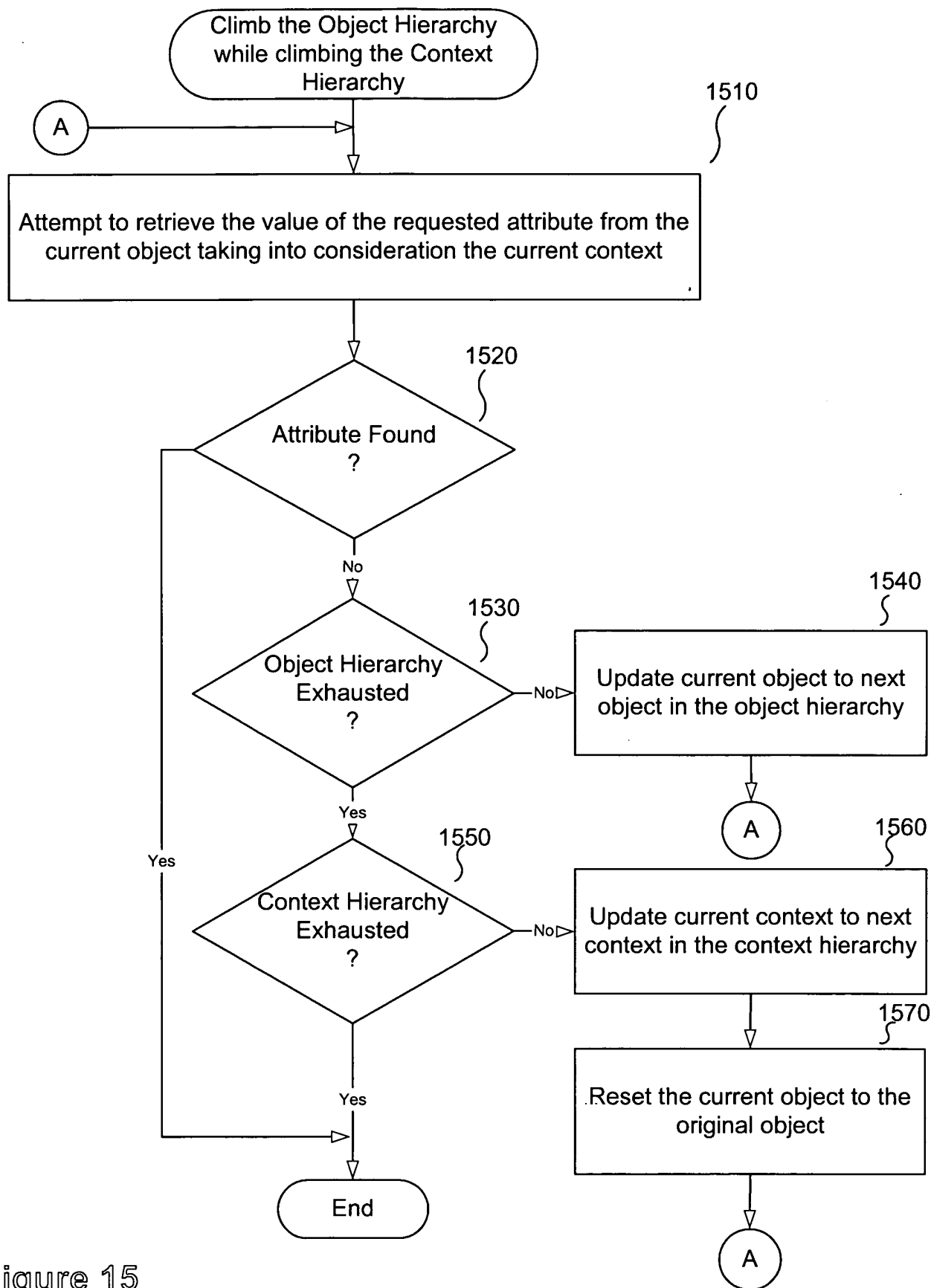


Figure 15

09826219.082701

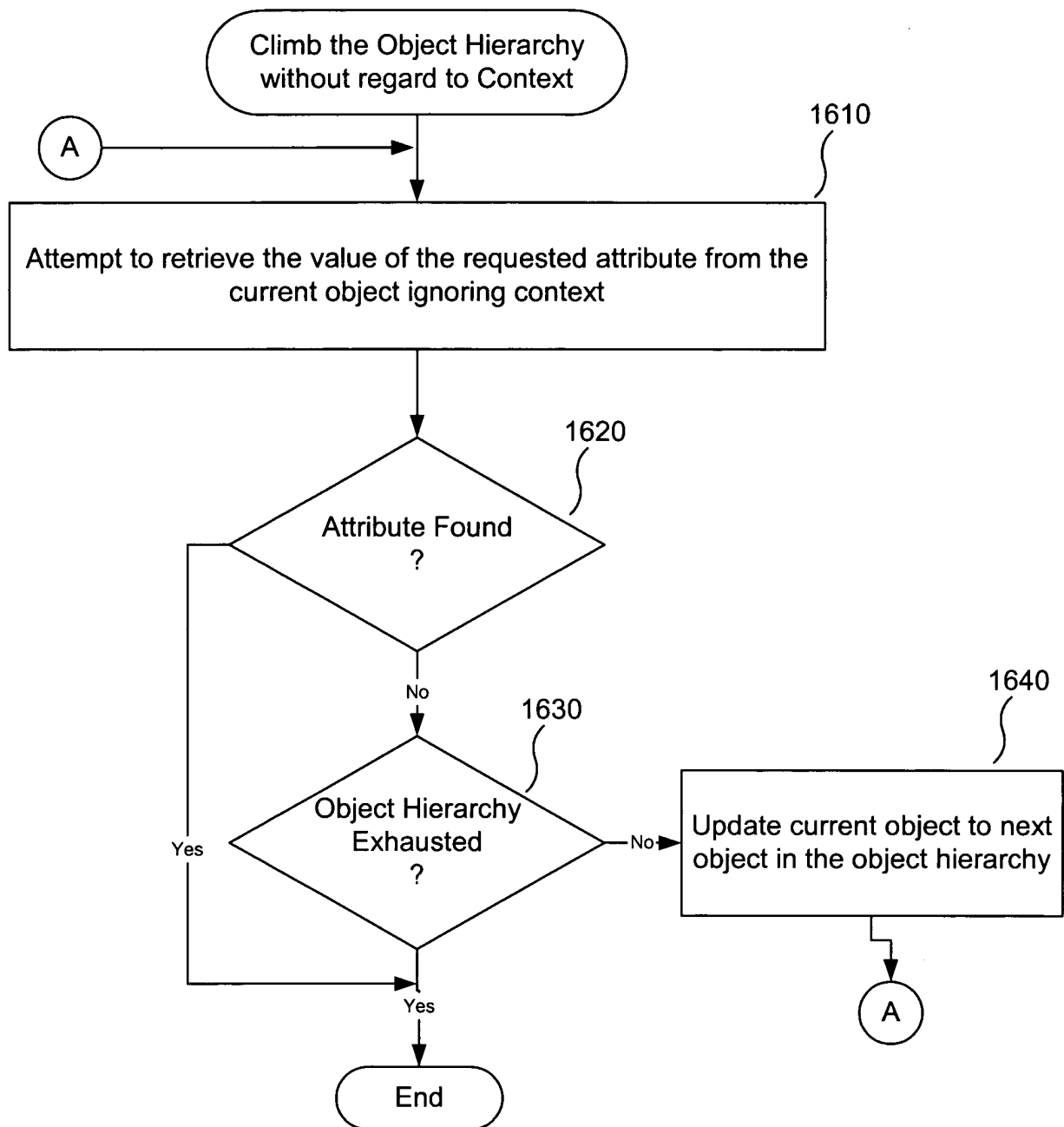


Figure 16

098619-082701

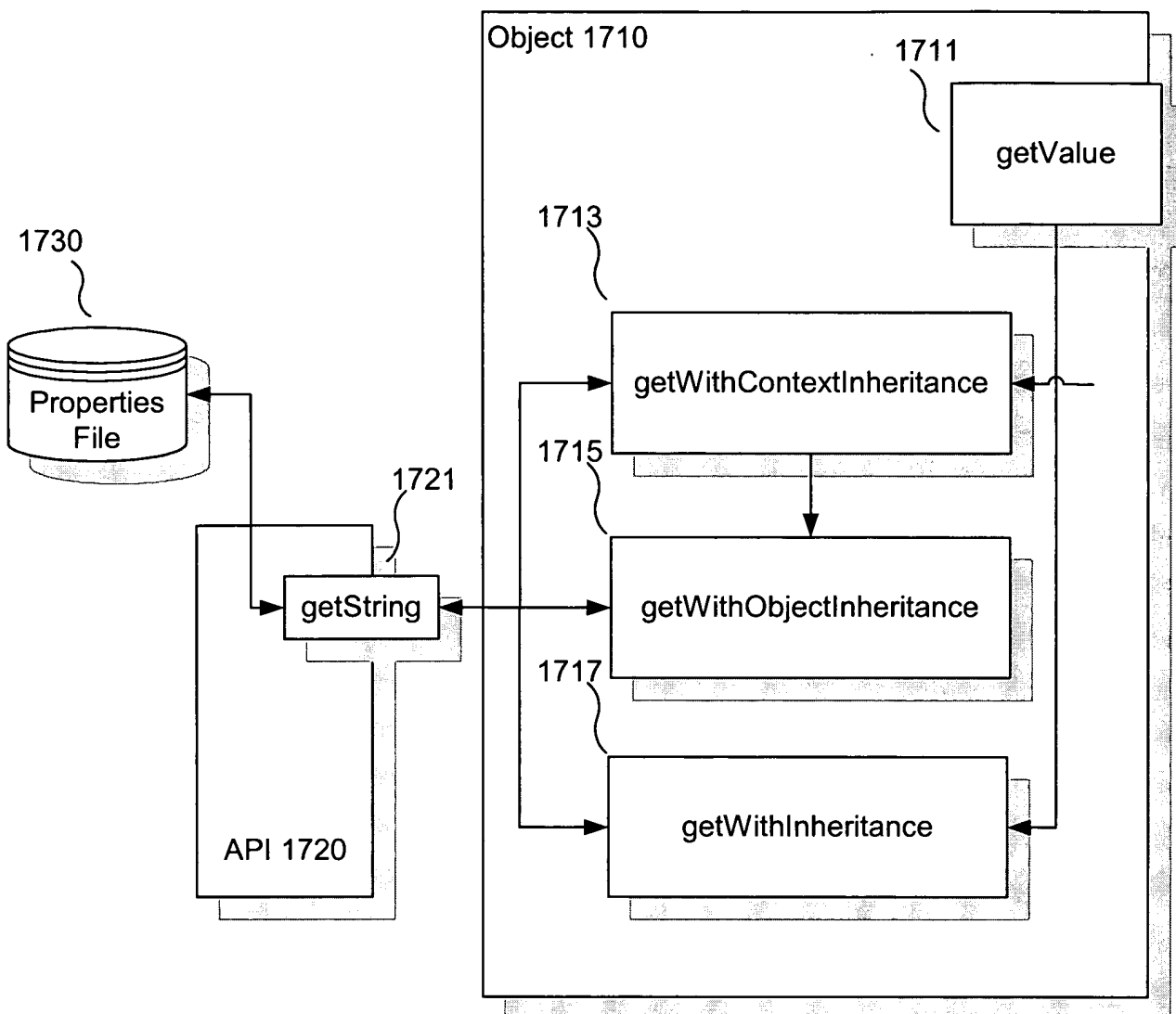


Figure 17


```
1.  String getValue( String attrib ) {
2.
3.      // get the container name of this object
4.      String context = getContextName();
5.
6.      // get the name of the class of this executing object
7.      String object = getClass().getName();
8.
9.      // check to see if within this context a value is defined for the
10.     // specified attribute for objects of the currently executing object's
11.     // type, or for any of its parent objects
12.
13.     String result = getWithContextInheritance( context, object, attrib );
14.
15.     if (result == null) { // if it's null it didn't find it.
16.
17.         // now try to find the value of the attribute of the object
18.         // by ignoring the context
19.
20.         // this will look for the attribute in all objects that are extended by
21.         // the current object
22.
23.         result = getWithInheritance( object, attrib );
24.
25.     } // end if
26.
27.     return result;
28.
29. } // end getValue
```

Figure 18

```

1.  String getWithContextInheritance(
2.      String context, String object, String attrib) {
3.
4.      // check if the attrib is found in the current context by climbing up
5.      // the object hierarchy if necessary
6.      String result = getWithObjInheritance( context, object, attrib );
7.
8.      if (result == null) { //didn't find it in the current context
9.          // check if this context extends another context
10.         context = ResourceBundle.getString( context + ".extends" );
11.
12.         //context now contains the name of the parent context, if any
13.         if (context != null) // found a parent context
14.             // this is a recursive call that looks for the attribute
15.             // and climbs up the context hierarchy
16.             result = getWithContextInheritance( context, object, attrib );
17.
18.         } //end if
19.
20.         return result;
21.
22.     } // end getWithContextInheritance

```

Figure 19

```

1.  String getWithObjInheritance( String context, String object, String attrib ) {
2.      // concatenate context, ".", the object, ".", and the attrib
3.      String lookFor = context + "." + object + "." + attrib;
4.
5.      // check if the combined attribute is found in the resource bundle
6.      String result = resourceBundle.getString( lookFor );
7.
8.      if (result == null) { // didn't find it in the current object
9.          // check if object extends another object, meaning
10.         // it has a parent.
11.         object = resourceBundle.getString( object + ".extends" );
12.
13.         //object now contains the name of the parent, if any
14.         if (object != null) // found a parent object
15.             // this is a recursive call that climbs up the
16.             // object's hierarchy while keeping the context unchanged.
17.             result = getWithObjInheritance( context, object, attrib );
18.
19.     } // end if
20.
21.     return result;
22.
23. } // end getWithObjInheritance

```

Figure 20

```
1. String getWithInheritance( String object, String attrib ) {
2.     // concatenate the object, ".", and the attrib
3.     String lookFor = object + "." + attrib;
4.
5.     // check if the combined attribute is found in the resource bundle
6.     String result = resourceBundle.getString( lookFor );
7.     if (result == null) { //didn't find it in the current object
8.         // check if object extends another object, meaning
9.         // it has a parent.
10.        object = resourceBundle.getString( object + ".extends" );
11.
12.        // object now contains the name of the parent, if any
13.        if (object != null) // found a parent object
14.            // this is a recursive call. Look for the attribute
15.            // in the parent or its parents
16.            result = getWithInheritance( object, attrib );
17.
18.    } // end if
19.
20.    return result;
21.
22. } // end getWithInheritance
```

Figure 21